```
from a0_items import *


'''
# initializing list
test_list = [1, 5, 6, 7, 4, 2323]

# printing original list
print("The original list is : " + str(test_list))

# using list indexing
# to get last element of list
res = [test_list[-1]]

# printing result
print("The last element of list are : " + str(res))
'''


##########################################################################################################################
#### Multi-processing
##########################################################################################################################

import multiprocessing
from multiprocessing import Process

print("Number of cpu : ", multiprocessing.cpu_count())



def fun_0():

     for x in range(100):
        print ("Shane")


def fun_1():

    for x in range(500):
       print ("Lucy")


def fun_2():

    for x in range(500):
       print ("ALEX")


def test123():

     p0 = Process(target=fun_0)
     p0.start()

     p1 = Process(target=fun_1)
     p1.start()

     p2 = Process(target=fun_2)
     p2.start()


'''
#must run multi-proce using if __name__ == '__main__'
if __name__ == '__main__':
    test123()
'''


###########################################################################################################################
#### Concurrent futures
###########################################################################################################################


from concurrent.futures import ThreadPoolExecutor, wait
import concurrent.futures
import time



Num1 = [1, 3, 4, 8]
Num2 = [8, 1, 3, 7]
Words1 = ['hey', 'yo', 'blue', 'no']
Words2 = ['Ho' , 'Ming', 'Peter', 'John']



def print_sth (var1, var2, var3, var4):

   print (var1 + var2, var3, var4)



# https://www.youtube.com/watch?v=IEEhzQoKtQU
# https://www.packetswitch.co.uk/what-is-concurrent-futures-and-how-can-it-boost-your-python-performance/
# https://superfastpython.com/processpoolexecutor-common-errors/
def concur_0_to_3 ():

   with concurrent.futures.ProcessPoolExecutor() as executor:

      executor.map(print_sth, Num1[2:4], Num2[2:4], Words1[2:4], Words2[2:4])

   t2 = time.perf_counter()
   print (f' finished in {t2} seconds')


#concur_0_to_3 ()




#########################################################################################################################
# Concurrent inside concurrent
###########################################################################################################################

LIST_a = ['a1', 'a2', 'a3','a4', 'a5']
LIST_b = ['b1', 'b2', 'b3','b4', 'b5']
LIST_c = ['c1', 'c2', 'c3','c4', 'c5']

def func_a (var_a):

   print (var_a)


def func_b (var_b):

```

```python
    print (var_b)


def func_c (var_c):

    print (var_c)



def CF_a ():

    with concurrent.futures.ProcessPoolExecutor() as executor:

            executor.map(func_a, LIST_a)

    t2 = time.perf_counter()
    print (f' finished in {t2} seconds')



def CF_b ():

    with concurrent.futures.ProcessPoolExecutor() as executor:

            executor.map(func_b, LIST_b)

    t2 = time.perf_counter()
    print (f' finished in {t2} seconds')



def CF_c ():

    with concurrent.futures.ProcessPoolExecutor() as executor:

            executor.map(func_c, LIST_c)

    t2 = time.perf_counter()
    print (f' finished in {t2} seconds')



def CF_sequence ():

    CF_a ()
    CF_b ()
    CF_c ()



def CF_ALL ():

    with ThreadPoolExecutor(3) as ex:
        futures = []
        futures.append(ex.submit(CF_a))
        futures.append(ex.submit(CF_b))
        futures.append(ex.submit(CF_c))



#must run concurrency using if __name__ == '__main__'
if __name__ == '__main__':

    CF_a()
```